

## Probeklausur Angewandte Informatik

Name	
Vorname	
Matrikelnummer	
Studiengang	

Hinweise:

1. Bearbeitungszeit: 90 Minuten (aber Klausur 120 min!)
2. **Zugelassene Hilfsmittel: keine.** Verstauen Sie Handy, Laptop und Uhr sowie alle weiteren elektronischen Geräte im ausgeschalteten Zustand in Ihrer Tasche. Wird ein elektronisches Gerät in Griffweite oder am Körper gefunden, zählt das als Täuschungsversuch.
3. Legen Sie Ihren Personalausweis und Studychip offen auf den Tisch.
4. Schreiben Sie leserlich.
5. Es sind Kugelschreiber, Füller oder Bleistift in allen Farben erlaubt, nur rot ist verboten.
6. Bitte wenden Sie sich an die Aufsicht, wenn Sie zusätzliche Blätter benötigen.
7. Das Lösen der Tackernadeln ist verboten.
8. Zum Bestehen der Klausur benötigen Sie mindestens 50 von 100 Punkten. Die erreichbaren Punkte stehen neben der Aufgabe.

Mit meiner Unterschrift bestätige ich,

1. dass ich meine Klausur selbständig angefertigt und keine anderen als die zugelassenen Hilfsmittel verwendet habe,
2. dass ich mich gesund und prüfungsfähig fühle. Mir ist bekannt, dass mit dem Erhalt der Aufgabenstellung die Klausur als angetreten gilt und bewertet wird.

---

Datum, Unterschrift

# 1 Theorie der Informatik

**Aufgabe 1.1.** Software ist im Gegensatz zu Hardware nicht eindeutig definiert. Die aktuelle ISO/IEC-Norm 24765 enthält für Software drei leicht unterschiedliche Definitionen, wir verwenden die umfangreichste Definition. Ergänzen Sie in der folgenden Definition für Software die korrekten Fachbegriffe:

Software sind Programme und ggf. die zugehörige Dokumentation und weitere Daten, die zum Betrieb eines Computers notwendig sind.

**Aufgabe 1.2.** Ordnen Sie die folgenden Objekte der passenden Hardware-Kategorie zu. Setzen Sie dazu das Kreuz in der Spalte mit der Kategorie, in die das Objekt gehört. Setzen Sie ein Kreuz in der Spalte „nichts davon“, wenn keine der Hardware-Kategorien Eingabe, Verarbeitung oder Ausgabe korrekt ist.

Objekt	Eingabe	Verarbeitung	Ausgabe	nichts davon
Beamer			X	
Compiler				X
Grafikprozessor (GPU)		X		
Interpreter				X
Kamera (Webcam)	X			
Kopfhörer			X	
Maus	X			
Mikrofon	X			
Monitor			X	
Prozessor (CPU)		X		
Tastatur	X			
Windows 10				X

**Aufgabe 1.3.** Beschreiben Sie in einem oder mehreren Sätzen, was in der Informatik das EVA-Prinzip bedeutet:

*Das EVA-Prinzip ist in der Informatik das grundlegende Prinzip der Datenverarbeitung. Die Buchstaben E, V, A stehen dabei als Abkürzung für die Begriffe Eingabe, Verarbeitung, Ausgabe.*

**Aufgabe 1.4.** Bitte ergänzen Sie in der folgenden Wahrheitstabelle das Ergebnis der logischen Operation NICHT:

NICHT	wahr	<i>falsch</i>
NICHT	falsch	<i>wahr</i>

**Aufgabe 1.5.** Bitte ergänzen Sie in der folgenden Wahrheitstabelle das Ergebnis der logischen Operation ODER:

falsch	ODER	falsch	<i>falsch</i>
falsch	ODER	wahr	<i>wahr</i>
wahr	ODER	falsch	<i>wahr</i>
wahr	ODER	wahr	<i>wahr</i>

## 2 Lesen von Code

**Aufgabe 2.1.** Was gibt der Python oder MATLAB-Interpreter beim Durchlaufen des folgenden Programmes aus? Bitte lesen Sie den Code in Ihrer bevorzugten Programmiersprache und wählen Sie danach unter den vorgegebenen Ausgaben die korrekte Ausgabe aus.

MATLAB:

```
1 x = 5;
2 y = x + 3;
3 x = x - 1;
4 z = 10;
5 x = x + z;
6 fprintf('x: %g, y: %g, z: %g', x,y,z);
```

Python:

```
1 x = 5
2 y = x + 3
3 x = x - 1
4 z = 10
5 x = x + z
6 print(f'x: {x}, y: {y}, z: {z}')
```

Kreuzen Sie mit X Ihre Antwort an, verwenden Sie ■ um die Antwort zu löschen:

- ☒ x: 14, y: 8, z: 10
- ☐ x: 17, y: 8, z: 10
- ☐ 14, 8, 10
- ☐ 17, 8, 10
- ☐ Keine der bisher genannten Optionen stimmt.

**Aufgabe 2.2.** Was gibt der Python oder MATLAB-Interpreter beim Durchlaufen des folgenden Programmes aus? Bitte lesen Sie den Code in Ihrer bevorzugten Programmiersprache und wählen Sie danach unter den vorgegebenen Ausgaben die korrekte Ausgabe aus.

MATLAB:

```
1 disp('Nein. ')
2 funktion()
3
4 function funktion()
5     disp('Ja. ')
6 end
```

~> Ausgabe: Nein.

~> Aufruf funktion() führt zu Ausgabe Ja.

Python:

```
1 def funktion():
2     print('Ja. ')
3
4 print('Nein. ')
5 funktion()
```

~> wird übersprungen

~> Ausgabe: Nein.

~> Aufruf funktion() führt zu Ausgabe Ja.

Kreuzen Sie mit X Ihre Antwort an, verwenden Sie ■ um die Antwort zu löschen:

☒ Nein.  
Ja.

☐ Ja. Nein.

☐ Ja.  
Nein.

☐ Nein. Ja.

☐ Keine der bisher genannten Optionen stimmt.

**Aufgabe 2.3.** Was gibt der Python oder MATLAB-Interpreter beim Durchlaufen des folgenden Programmes aus? Bitte lesen Sie den Code in Ihrer bevorzugten Programmiersprache und wählen Sie danach unter den vorgegebenen Ausgaben die korrekte Ausgabe aus.

MATLAB:

```
1 a = 1;
2 while a < 3
3     a = a + 2;
4     disp(a^2);
5 end
6 disp(a)
```

$a = 1$   
 1. Schleifendurchgang:  
 $a = 1 + 2 = 3$   
 $\text{disp}(3^2) \leadsto \text{Ausgabe } 9$   
 da die Bedingung  $a < 3$  nicht erfüllt ist, gibt es keine weiteren Schleifendurchgänge  
 $\text{disp}(3) \leadsto \text{Ausgabe } 3$

Python:

```
1 a = 1
2 while a < 3:
3     a = a + 2
4     print(a**2)
5 print(a)
```

$a = 1$   
 1. Schleifendurchgang:  
 $a = 1 + 2 = 3$   
 $\text{print}(3^2) \leadsto \text{Ausgabe } 9$   
 da die Bedingung  $a < 3$  nicht erfüllt ist, gibt es keine weiteren Schleifendurchgänge  
 $\text{print}(3) \leadsto \text{Ausgabe } 3$

Kreuzen Sie mit X Ihre Antwort an, verwenden Sie ■ um die Antwort zu löschen:

☐ 1

3

☐ 1

9

☐ 3

9

☒ 9

3

☐ Keine der bisher genannten Optionen stimmt.

**Aufgabe 2.4.** Was gibt der Python oder MATLAB-Interpreter beim Durchlaufen des folgenden Programmes aus? Bitte lesen Sie den Code in Ihrer bevorzugten Programmiersprache und wählen Sie danach unter den vorgegebenen Ausgaben die korrekte Ausgabe aus.

MATLAB:

```
1 nummer_1 = 5;
2 if nummer_1 >= 6
3     nummer_2 = 3;
4 else
5     nummer_2 = 4;
6     x = nummer_2 * nummer_1 + 1;
7 end
8 disp(x)
```

*nummer\_1 = 5  
da  $5 \geq 6$  falsch ist, geht es mit else in Zeile 4 weiter  
else  
nummer\_2 = 4  
 $x = 4 \cdot 5 + 1 = 21$   
disp(21)  $\leadsto$  Ausgabe 21*

Python:

```
1 nummer_1 = 5
2 if nummer_1 >= 6:
3     nummer_2 = 3
4 else:
5     nummer_2 = 4
6     x = nummer_2 * nummer_1 + 1
7 print(x)
```

*nummer\_1 = 5  
da  $5 \geq 6$  falsch ist, geht es mit else in Zeile 4 weiter  
else  
nummer\_2 = 4  
 $x = 4 \cdot 5 + 1 = 21$   
print(21)  $\leadsto$  Ausgabe 21*

Kreuzen Sie mit X Ihre Antwort an, verwenden Sie ■ um die Antwort zu löschen:

☐ 16

☒ 21

☐ x

☐ nummer\_2 \* nummer\_1 + 1

☐ Keine der bisher genannten Optionen stimmt.

### 3 Programmieren

**Aufgabe 3.1.** Schreiben Sie ein Programm, mit dem das kleine Einmaleins geübt werden kann. Der Computer soll 10 Rechenaufgaben stellen, z.B.

Wieviel ist  $3 \times 4$ ?

und dann die Antwort des Benutzers oder der Benutzerin einlesen. Falls das Ergebnis richtig ist, soll der Computer schreiben: "Richtig!" und ansonsten "Falsch, das richtige Ergebnis wäre xx gewesen." Nach den 10 Aufgaben soll der Computer noch mitteilen, wieviel Prozent der Aufgaben richtig gelöst wurden.

```
NATLAB:
anzahl_richtig = 0;
for i = 1:10
    x = randi([1,10]);
    y = randi([1,10]);
    fprintf('Wieviel ist %d x %d ?\n', x,y);
    xy = input(' ');
    if xy == x * y
        disp('Richtig!');
        anzahl_richtig = anzahl_richtig + 1;
    else
        fprintf('Falsch, das richtige Ergebnis wäre %d gewesen.\n', x*y);
    end
end
fprintf('Es wurden %f Prozent der Aufgaben gelöst.\n', anzahl_richtig * 10);
```



## Python

```
import numpy as np
```

```
anzahl_richtig = 0
```

```
for i in range(10):
```

```
    x = np.random.randint(1, 11)
```

```
    y = np.random.randint(1, 11)
```

```
    xy = int(input(f'Wieviel ist {x} x {y}?. '))
```

```
    if xy == x * y:
```

```
        print('Richtig!')
```

```
        anzahl_richtig = anzahl_richtig + 1
```

```
    else:
```

```
        print(f'Falsch, das richtige Ergebnis wäre {xy} gewesen.')
```

```
print(f'Es wurden {anzahl_richtig * 10} Prozent der Aufgaben gelöst.')
```

## 4 Fehlersuche

**Aufgabe 4.1.** Suchen Sie in dem folgenden Programm die Fehler. Markieren Sie die Fehler, indem Sie den Fehler umkreisen. Schreiben Sie anschließend in der folgenden Tabelle die Zeilennummer, in der der Fehler auftritt, eine Begründung, warum dies ein Fehler ist, und zuletzt, wie die Zeile korrekt heißen müsste.

Da folgende Programm soll die Fakultät einer Zahl berechnen, die der Benutzer eingibt. Die Fakultät einer Zahl  $n$  wird mit  $n!$  bezeichnet und gemäß der Formel  $n! = n \cdot (n-1) \cdot \dots \cdot 2 \cdot 1$  berechnet.

```

1 # Python
2
3 # Eingabe
4 n = int(input('Bitte geben Sie n ein: '))
5
6 # Verarbeitung
7 for i in range(1, n+1):
8     ergebnis = ergebnis * i
9
10 # Ausgabe
11 print(f'Die Fakultät von {n} lautet {n}! = {ergebnis}.')
```

```

1 % MATLAB
2
3 % Eingabe
4 n = input('Bitte geben Sie n ein: ');
5
6 % Verarbeitung
7 for i = 1:n
8     ergebnis = ergebnis * i;
9 end
10
11 % Ausgabe
12 fprintf('Die Fakultät von %g lautet %g ! = %g, n, n, ergebnis);
```

Zeilennr.	Fehlerbegründung	Fehlerkorrektur
4	Anführungszeichen fehlt	input('Bitte geben Sie n ein: ')
8	Variablen ergebnis nicht initialisiert	in Zeile 5: ergebnis = 1
11	Platzhalter fehlt, f-String fehlt	print(f'Bitte
	MATLAB PYTHON	lautet '%g ! = %g.', n, n, ergebnis
		(Python) (Matlab)

**Aufgabe 4.2.** Suchen Sie in dem folgenden Programm die Fehler. Markieren Sie die Fehler, indem Sie den Fehler umkreisen. Schreiben Sie anschließend in der folgenden Tabelle die Zeilennummer, in der der Fehler auftritt, eine Begründung, warum dies ein Fehler ist, und zuletzt, wie die Zeile korrekt heißen müsste.

Das folgende Programm soll ermitteln, wie oft man einen Würfel werfen muss, bis zum ersten Mal eine 6 erscheint. Dazu wird eine Funktion implementiert, die so lange würfelt, bis eine 6 erschienen ist. Als Rückgabe gibt die Funktion die Anzahl der Würfe zurück. Bei dem anschließenden Hauptprogramm darf ein Benutzer eingeben, wie oft der Würfelversuch durchgeführt werden soll. Anschließend wird die Funktion so oft ausgeführt, wie vom Benutzer angegeben. Zuletzt wird der Mittelwert der Würfelversuche berechnet und ausgegeben.

```
1  # Python
2  import numpy as np
3
4  ## Funktion
5  def wuerfeln_bis_sechs():
6      anzahl_wuerfe = 1
7      wurf = np.random.randint()
8      while wurf == 6:
9          anzahl_wuerfe = anzahl_wuerfe + 1
10         wurf = np.random.randint(1, 7)
11
12     return anzahl_wuerfe
13
14
15 # Hauptprogramm
16 # Eingabe
17 anzahl_versuche = int(input('Wie viele Würfelversuche? '))
18
19 # Verarbeitung
20 summe_versuche = 0
21 for i in range(anzahl_versuche + 1):
22     summe_versuche = summe_versuche + wuerfeln_bis_sechs()
23
24 mittelwert = summe_versuche / anzahl_versuche
25
26 # Ausgabe
27 print(f'Durchschnitt: {mittelwert}')
```

*Argumente für randint()-Funktion fehlen  
Korrektur: wurf = np.random.randint(1,7)*

*semantischer Fehler, wurf != 6*

*ein Schleifendurchgang zuviel*

*syntaktischer Fehler: Doppelpunkt fehlt*

```

1 % MATLAB
2
3 % Hauptprogramm
4 % Eingabe
5 anzahl_versuche = input('Wie viele Würfelversuche? ');
6
7 % Verarbeitung
8 summe_versuche = 0;
9 for i = 1:anzahl_versuche
10     summe_versuche = summe_versuche + wuerfeln_bis_sechs();
11 end
12 mittelwert = summe_versuche / anzahl_versuche;
13
14 % Ausgabe
15 fprintf('Durchschnitt: %.1f', mittelwert);
16
17 %% Funktion
18 function anzahl_wuerfe = wuerfeln_bis_sechs()
19     anzahl_wuerfe = 1;
20     wurf = randi();
21     while wurf == 6
22         anzahl_wuerfe = anzahl_wuerfe + 1;
23         wurf = randi([1, 6]);
24
25     end
26 end

```

MATLAB

Zeilennr.	Fehlerbegründung	Fehlerkorrektur
11	syntaktischer Fehler, end fehlt	Zeile 11: end
20	Argumente der randi-Funktion fehlen	wurf = randi([1,6])
21	semantischer Fehler: es muss ungleich 6 sein	while wurf ~= 6

Python

7	Argumente der randint-Funktion fehlen	wurf = np.random.randint(1,7)
8	semantischer Fehler: es muss ungleich 6 sein	while wurf != 6:
21	syntaktischer Fehler: Doppelpunkt fehlt	for i in range(anzahl_versuche):
21	anzahl_versuche + 1 ist ein Durchgang zuviel	—  —

## 5 Wissenschaftliche Anwendungen

**Aufgabe 5.1.** In einer Klausur gibt es folgenden Notenspiegel

Note	1	2	3	4	5
Anzahl	3	8	11	4	0

1. Beschreiben Sie, welchen Diagrammtyp Sie zur Darstellung dieser Daten wählen würden.

Balkendiagramm (Säulendiagramm)

2. Begründen Sie, warum Sie diesen Diagrammtyp wählen würden.

Die Daten sollten durch ein Balkendiagramm visualisiert werden, weil es sich um diskrete Daten (Kategorien) handelt.

3. Schreiben Sie anschließend den MATLAB- oder Python-Code, der notwendig wäre, um diese Daten zu visualisieren. Beschriften Sie auch die Achsen und setzen Sie einen Titel.

MATLAB

```
x = [1, 2, 3, 4, 5];  
y = [3, 8, 11, 4, 0];  
figure();  
bar(x,y);  
xlabel('Note');  
ylabel('Anzahl');  
title('Notenspiegel');
```

```
import matplotlib.pyplot as plt  
x = [1, 2, 3, 4, 5]  
y = [3, 8, 11, 4, 0]  
plt.figure()  
plt.bar(x,y)  
plt.xlabel('Note')  
plt.ylabel('Anzahl')  
plt.title('Notenspiegel')
```

PYTHON

**Aufgabe 5.2.** Gegeben sind die folgenden Daten: Gegeben sind die folgenden Daten:

Jahr	2005	2010	2015	2020
Messwert	29602	33343	35496	34807

1. Zeichnen Sie per Hand die Daten in ein Diagramm.
2. Wählen Sie eine Modellfunktion, die die Daten möglichst gut interpoliert, und skizzieren Sie die Modellfunktion ebenfalls in dem Diagramm.
3. Geben Sie die mathematische Funktionsvorschrift der Modellfunktion an.
4. Geben Sie die Parameter der Modellfunktion an.
5. Schreiben Sie anschließend den MATLAB- oder Python-Code, der notwendig wäre, um die Parameter der von Ihnen gewählten Modellfunktion zu bestimmen.

